

CSCI 331

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	bplustreemk4 Class Reference . . . . .	5
3.1.1	Constructor & Destructor Documentation . . . . .	5
3.1.1.1	bplustreemk4() [1/2] . . . . .	5
3.1.1.2	bplustreemk4() [2/2] . . . . .	6
3.1.2	Member Function Documentation . . . . .	7
3.1.2.1	cleanUp() . . . . .	7
3.1.2.2	processFile() . . . . .	7
3.2	indexB Struct Reference . . . . .	8
3.2.1	Constructor & Destructor Documentation . . . . .	8
3.2.1.1	indexB() [1/2] . . . . .	9
3.2.1.2	indexB() [2/2] . . . . .	9
3.2.2	Member Function Documentation . . . . .	9
3.2.2.1	setNext() . . . . .	9
3.2.2.2	writeToFile() . . . . .	9
3.2.3	Member Data Documentation . . . . .	10
3.2.3.1	nextPointer . . . . .	10
3.2.3.2	prevPointer . . . . .	10

3.2.3.3	sequence	10
3.3	kp Struct Reference	10
3.3.1	Constructor & Destructor Documentation	10
3.3.1.1	kp()	11
3.3.2	Member Function Documentation	11
3.3.2.1	getValue()	11
3.3.2.2	setValue()	11
3.3.3	Member Data Documentation	11
3.3.3.1	key	11
3.3.3.2	value	12
3.4	sequenceB Struct Reference	12
3.4.1	Constructor & Destructor Documentation	12
3.4.1.1	sequenceB() [1/2]	12
3.4.1.2	sequenceB() [2/2]	12
3.4.2	Member Function Documentation	13
3.4.2.1	setNext()	13
3.4.2.2	writeToFile()	13
3.4.3	Member Data Documentation	13
3.4.3.1	nextPointer	13
3.4.3.2	prevPointer	14
3.4.3.3	sequence	14
<b>4</b>	<b>File Documentation</b>	<b>15</b>
4.1	bplustreemk4.cpp File Reference	15
4.2	bplustreemk4.h File Reference	15
4.3	main.cpp File Reference	15
4.3.1	Function Documentation	16
4.3.1.1	main()	16
4.3.2	Variable Documentation	16
4.3.2.1	dir	16
<b>Index</b>		<b>17</b>

# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">bplustreemk4</a>	5
<a href="#">indexB</a>	8
<a href="#">kp</a>	10
<a href="#">sequenceB</a>	12



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">bplustreemk4.cpp</a>	15
<a href="#">bplustreemk4.h</a>	15
<a href="#">main.cpp</a>	15





## Class Documentation

```
#include <bplustreemk4.h>
```

- `bplustreemk4 ()`
- `bplustreemk4 (string directory)`
- `void processFile ()`
- `void cleanUp ()`

### 3.1.1.1 bplustreemk4() [1/2]

```
bplustreemk4::bplustreemk4 ( )
```

```

107                                     {
108
109 }

```

## 3.1.1.2 bplustreemk4() [2/2]

```

bplustreemk4::bplustreemk4 (
    string directory )

9
10
11 //ofstream out;
12 fstream myfile;
13 this->dir = dir;
14 string header;
15 string headerout;
16 myfile.open(dir, ios::in);
17 myfile >> this->headerSize;
18 getline(myfile, header);
19 cout << header << endl;
20 this->header = header;
21 int fieldcount = 0;
22 int count = 0;
23 //a "different" way to read data in because it wasnt working
24 for (int i = 1; i < this->headerSize - 2; i++) {
25     if (header.at(i) == '|') {
26         cout << "encounter a field marker" << endl;
27         fieldcount++;
28         string temp;
29         for (int x = i - count; x < i; x++) {
30             temp.push_back(header.at(x));
31         }
32         if (fieldcount == 1) {
33             this->insertFlag = stoi(temp);
34         }
35         if (fieldcount == 2) {
36             this->totalNumRecords = stoi(temp);
37         }
38         if (fieldcount == 3) {
39             this->sizeofRecords = stoi(temp)-1;
40         }
41         if (fieldcount == 4) {
42             this->fieldsPerRecord = stoi(temp);
43         }
44         if (fieldcount == 5) {
45             cout << "temp aka what ever we have deemed a field " << temp;
46             /*vector<string> temp2;
47             for (int y = 0; y < temp.size(); y++) {
48
49             }*/
49             vector<string> temp2;
50             string buffer;
51             stringstream ss(buffer);
52             for (int y = 0; y < temp.size(); y++) {
53                 ss.clear();
54                 ss << temp.at(y);
55                 //cout << "Went into the string stream" << temp.at(y) << endl;
56                 ss >> buffer;
57                 //cout << "THIS IS WHAT IS IN THE BUFFER " << buffer << endl;
58                 if (buffer != "\"" || buffer != "(" || buffer != ")") {
59                     temp2.push_back(buffer);
60                 }
61             }
62             this->sizePerfield = temp2;
63         }
64         if (fieldcount == 6) {
65             vector<char> temp3;
66             for (int y = 0; y < temp.size(); y++) {
67                 if (temp.at(y) != '"' || temp.at(y) != '(' || temp.at(y) != ')') {
68                     temp3.push_back(temp.at(y));
69                 }
70             }
71             this->labelPerfield = temp3;
72         }
73         if (fieldcount == 7) {
74             this->sortField = stoi(temp);
75         }
76         if (fieldcount == 8) {
77             this->description = temp;
78         }
79         count = 0;
80         i++;
81     }
82     count++;
83 }
84 myfile.close();

```

```

87     this->rbnRoot = this->header.size();
88     //debugging
89     cout << this->headerSize << endl;
90     cout << this->totalNumRecords << endl;
91     cout << this->description << endl;
92
93     /*for (int i = 0 ; i < this->sizePerfield.size(); i++) {
94         cout << this->sizePerfield.at(i);
95     }
96     cout << endl;
97     for (int i = 0; i < this->labelPerfield.size(); i++) {
98         cout << this->labelPerfield.at(i);
99     }*/
100    //out.open("btree.dat");
101    //out << header;
102    //out.close();
103    cleanUp();
104    processFile();
105 }
106 }

```

### 3.1.2 Member Function Documentation

#### 3.1.2.1 cleanUp()

void bplustreemk4::cleanUp ( )

```

160     {
161         vector<string> tempf;
162         string temp;
163         for (int i = 0; i < this->sizePerfield.size(); i++) {
164             if (this->sizePerfield.at(i) != "\"" && this->sizePerfield.at(i) != "(" && this->sizePerfield.at(i)
165                 != ")") {
166                 cout << this->sizePerfield.at(i);
167                 tempf.push_back(this->sizePerfield.at(i));
168             }
169             this->sizePerfield = tempf;
170             cout << endl;
171             for (int i = 0; i < this->labelPerfield.size(); i++) {
172                 cout << this->labelPerfield.at(i);
173             }
174 }

```

#### 3.1.2.2 processFile()

void bplustreemk4::processFile ( )

```

110     {
111         ofstream outfile;
112         fstream myfile;
113         myfile.open(this->dir);
114         outfile.open("btree.dat");
115         outfile << this->header;
116         sequenceB sb;
117         indexB ib;
118         //pair temp;
119         int nextP=0;
120         int preP=0;
121         this->totalSequenceBlocks = 0;
122         string tempLine;
123         while (getline(myfile, tempLine)) {
124             string zip;
125             if (tempLine.size() != this->sizeofRecords) {

```

```

126         cout << "Line that was not big enough dedcted." << endl;
127         continue;
128     }
129     else {
130         cout << "The line is of valid lengthh " << endl;
131         zip = tempLine.substr(0, 5);
132         cout << " The zip is " << zip << endl;
133         sb.nextPointer = this->headerSize + (this->blockSize * this->totalSequenceBlocks);
134         cout << "Next pointer is " << sb.nextPointer << endl;
135         cout << "pushing back this : " << tempLine << endl;
136         sb.segeunce.push_back(tempLine);
137         cout << " SEQUENCE SIZE BEFORE ADDING " << sb.segeunce.size() << endl;
138         ib.segeunce.push_back(kp(zip, (this->headerSize + this->totalSequenceBlocks * 249)));
139         if (sb.segeunce.size() == 3) {
140             this->totalSequenceBlocks++;
141             //cout << "this is what is in the sequence block " << sb.writeToFile() << endl;
142             outfile << sb.writeToFile();
143             sb.segeunce.clear();
144             continue;
145         }
146         if (ib.segeunce.size() == this->indexBlockSize-1) {
147             this->totalIndexBlock++;
148             cout << "this is what is in the Index block " << ib.writeToFile() << endl;
149             outfile << ib.writeToFile();
150             ib.segeunce.clear();
151             continue;
152         }
153         //cout << "The sequence size after adding " << sb.segeunce.size() << endl;
154     }
155 }
156 outfile.close();
157 myfile.close();
158 cout << this->sizeofRecords << endl;
159 }

```

The documentation for this class was generated from the following files:

- [bplustreemk4.h](#)
- [bplustreemk4.cpp](#)

## 3.2 indexB Struct Reference

```
#include <bplustreemk4.h>
```

### Public Member Functions

- [indexB](#) ()
- [indexB](#) (int prev, [kp](#) item)
- void [setNext](#) (int next)
- string [writeToFile](#) ()

### Public Attributes

- int [nextPointer](#)
- int [prevPointer](#)
- vector< [kp](#) > [segeunce](#)

#### 3.2.1 Constructor & Destructor Documentation

### 3.2.1.1 indexB() [1/2]

```
indexB::indexB ( ) [inline]
```

```
70     {
71
72     }
```

### 3.2.1.2 indexB() [2/2]

```
indexB::indexB (
    int prev,
    kp item ) [inline]
```

```
73     {
74         this->prevPointer = prev;
75         this->segeunce.push_back(item);
76     }
```

## 3.2.2 Member Function Documentation

### 3.2.2.1 setNext()

```
void indexB::setNext (
    int next ) [inline]
```

```
77     {
78         this->nextPointer = next;
79     }
```

### 3.2.2.2 writeToFile()

```
string indexB::writeToFile ( ) [inline]
```

```
80     {
81         string temp;
82         stringstream ss(temp);
83         temp.push_back('i');
84         temp.push_back('|');
85         for (int i = 0; i < segeunce.size(); i++) {
86             ss >> temp;
87         }
88         temp.push_back('|');
89         return temp;
90     }
```

### 3.2.3 Member Data Documentation

#### 3.2.3.1 nextPointer

```
int indexB::nextPointer
```

#### 3.2.3.2 prevPointer

```
int indexB::prevPointer
```

#### 3.2.3.3 sequeunce

```
vector<kp> indexB::sequeunce
```

The documentation for this struct was generated from the following file:

- [bplustreemk4.h](#)

## 3.3 kp Struct Reference

```
#include <bplustreemk4.h>
```

### Public Member Functions

- [kp](#) (string nkey, int [value](#))
- void [setValue](#) (int nValue)
- int [getValue](#) ()

### Public Attributes

- string [key](#)
- int [value](#)

### 3.3.1 Constructor & Destructor Documentation

### 3.3.1.1 kp()

```
kp::kp (
    string nkey,
    int value ) [inline]

10
11         this->key = nkey;
12         this->value = value;
13     }
```

## 3.3.2 Member Function Documentation

### 3.3.2.1 getValue()

```
int kp::getValue ( ) [inline]

17
18         return this->value;
19     }
```

### 3.3.2.2 setValue()

```
void kp::setValue (
    int nValue ) [inline]

14
15         this->value = nValue;
16     }
```

## 3.3.3 Member Data Documentation

### 3.3.3.1 key

```
string kp::key
```

### 3.3.3.2 value

```
int kp::value
```

The documentation for this struct was generated from the following file:

- [bplustreemk4.h](#)

## 3.4 sequenceB Struct Reference

```
#include <bplustreemk4.h>
```

### Public Member Functions

- [sequenceB](#) ()
- [sequenceB](#) (int prev, string item)
- void [setNext](#) (int next)
- string [writeToFile](#) ()

### Public Attributes

- int [nextPointer](#)
- int [prevPointer](#)
- vector< string > [sequeunce](#)

### 3.4.1 Constructor & Destructor Documentation

#### 3.4.1.1 sequenceB() [1/2]

```
sequenceB::sequenceB ( ) [inline]
```

```
27         {
28
29     }
```

#### 3.4.1.2 sequenceB() [2/2]

```
sequenceB::sequenceB (
    int prev,
    string item ) [inline]
```

```
30         {
31     this->prevPointer = prev;
32     this->sequeunce.push_back(item);
33 }
```



## 3.4.2 Member Function Documentation

### 3.4.2.1 setNext()

```
void sequenceB::setNext (
    int next ) [inline]
```

```
34         {
35         this->nextPointer = next;
36     }
```

### 3.4.2.2 writeToFile()

```
string sequenceB::writeToFile ( ) [inline]
```

```
37         {
38         string temp;
39         stringstream ss(temp);
40         temp.push_back('s');
41         //temp.push_back('|');
42         //temp.push_back(prevPointer);
43         temp.push_back('|');
44         for (int i = 0; i < sequeunce.size(); i++) {
45             for (int x = 0; x < sequeunce.at(i).size(); x++) {
46                 temp.push_back(sequeunce.at(i).at(x));
47             }
48             //ss << sequeunce.at(i);
49             //cout << "What is going into the sequence " << sequeunce.at(i) << endl;
50             //ss >> temp;
51         }
52         //cout << "THE LAST ITEM IN THE SEQUENCE " << sequeunce.at(2)<<endl;
53         //cout << "THE SIZE OF THE SEQUENCE BLOCK " << sequeunce.size() << endl;
54         temp.push_back('|');
55         temp.push_back(nextPointer);
56         temp.push_back('|');
57         //cout << "WHAT IS BEING RETURNED TO BE PRINTED " << temp << endl;
58         //for (int i = 0; i < sequeunce.size(); i++) {
59         //    cout << "IN THE SEQUENCE " << sequeunce.at(i) << endl;
60         //}
61         return temp;
62     }
```

## 3.4.3 Member Data Documentation

### 3.4.3.1 nextPointer

```
int sequenceB::nextPointer
```

### 3.4.3.2 prevPointer

```
int sequenceB::prevPointer
```

### 3.4.3.3 sequeunce

```
vector<string> sequenceB::sequeunce
```

The documentation for this struct was generated from the following file:

- [bplustreemk4.h](#)

## Chapter 4

# File Documentation

### 4.1 bplustreemk4.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <vector>
#include "bplustreemk4.h"
```

### 4.2 bplustreemk4.h File Reference

```
#include <string>
#include <sstream>
#include <vector>
```

#### Classes

- struct [kp](#)
- struct [sequenceB](#)
- struct [indexB](#)
- class [bplustreemk4](#)

### 4.3 main.cpp File Reference

```
#include <string>
#include <iostream>
#include <sstream>
#include <fstream>
#include "bplustreemk4.h"
```

## Functions

- int `main` ()

## Variables

- string `dir` = "C:\\Users\\reali\\Desktop\\CSCI\\us\_postal\_codesn+.txt"

### 4.3.1 Function Documentation

#### 4.3.1.1 `main()`

```
int main ( )

9      {
10     /*Due to the lack of time this is implemented staticly*/
11     bplustreemk4 test = bplustreemk4(dir);
12     return 0;
13 }
```

### 4.3.2 Variable Documentation

#### 4.3.2.1 `dir`

```
string dir = "C:\\Users\\reali\\Desktop\\CSCI\\us_postal_codesn+.txt"
```

# Index

- bplustreemk4, 5
  - bplustreemk4, 5
  - cleanUp, 7
  - processFile, 7
- bplustreemk4.cpp, 15
- bplustreemk4.h, 15
- cleanUp
  - bplustreemk4, 7
- dir
  - main.cpp, 16
- getValue
  - kp, 11
- indexB, 8
  - indexB, 8, 9
  - nextPointer, 10
  - prevPointer, 10
  - sequence, 10
  - setNext, 9
  - writeToFile, 9
- key
  - kp, 11
- kp, 10
  - getValue, 11
  - key, 11
  - kp, 10
  - setValue, 11
  - value, 11
- main
  - main.cpp, 16
- main.cpp, 15
  - dir, 16
  - main, 16
- nextPointer
  - indexB, 10
  - sequenceB, 13
- prevPointer
  - indexB, 10
  - sequenceB, 13
- processFile
  - bplustreemk4, 7
- sequence
  - indexB, 10
- sequenceB, 14
- sequenceB, 12
  - nextPointer, 13
  - prevPointer, 13
  - sequence, 14
  - sequenceB, 12
  - setNext, 13
  - writeToFile, 13
- setNext
  - indexB, 9
  - sequenceB, 13
- setValue
  - kp, 11
- value
  - kp, 11
- writeToFile
  - indexB, 9
  - sequenceB, 13